Subject: [SSGM 4.0 Plugin] TaggingSystem
Posted by roszek on Sat, 27 Oct 2012 14:33:58 GMT
View Forum Message <> Reply to Message

This is a very simple tagging plugin that I made for the map/mod test server that I use.

It seems to do what I want it to but if there is something wrong with the code I'm sure someone will let me know. keep in mind I know very little of making ssgm plugins.

How it works:

Basically it keeps track of scores over a period of a month and tags a player based on the scores they have accumulated; the tags are updated when a player joins or when a new level loads.

The scores and current month are stored in a file in the server folder called scores.txt; this file gets created if it does not exist.

At the beginning of a new month all data gets deleted.

The scores (which cause the tag updates), and the tag titles are read from the ssgm.ini file so that they can easily be changed.

The source code:

Stack.h
I created a custom stack class because I had trouble working with the engine_vector.
Toggle Code

```
#ifndef STACK_H
#define STACK_H
template <class T>
class Stack
{
private:
 struct node
 {
    T data;
    node * next;
 }*top;

public:
 Stack();
       ~Stack();

 void Create();
 void Push(T);
 void Pop();
 bool isEmpty();
```

```
        void Destroy();
 T Peek_Top();
 int Count();
 T Peek_At(int);
 float Get_Score(const char*);
 bool Set_Score(const char*, float);
 bool Find_Name(const char*);
};

template <class T>
Stack<T>::Stack(){Create();}

template <class T>
Stack<T>::~Stack(){Destroy();}

template <class T>
void Stack<T>::Push(T element)
{
   node *newnode;
   newnode = new node;
   newnode -> data = element;
   newnode -> next = top;
   top = newnode;
}

template <class T>
void Stack<T>::Pop()
{
   if(isEmpty())
      return;
   else
   {
      node *ptr;
      ptr = top;
      top = ptr -> next;
      delete ptr;
   }
}

template < class T >
bool Stack<T>::isEmpty()
{
   if(top == NULL)
      return true;
   else
      return false;
}
```

```
template < class T >
T Stack<T>::Peek_Top()
{
   return(top -> data);
}

template < class T >
T Stack<T>::Peek_At(int iter)
{
   node *ptr;
   ptr = top;
   for(int i = 0; i != iter; i++)
      ptr = ptr -> next;

   return(ptr -> data);
}

template < class T >
void Stack<T>::Create()
{
   top = NULL;
}

template <class T>
void Stack<T>::Destroy()
{
   while(!isEmpty()){Pop();}
}

template < class T >
int  Stack<T>::Count()
{
    int count = 0;
    node *ptr;
    ptr = top;
    while(ptr != NULL)
    {
       count = count + 1;
       ptr = ptr -> next;
    }
    return count;
}

template < class T >
float Stack<T>::Get_Score(const char * player)
{
     node *ptr;
     ptr = top;
```

```
    if(isEmpty())
      return ( 0 );
    while(ptr -> data.name != player)
    {
      ptr = ptr -> next;
      if(ptr == NULL)
         return ( 0 );
    }
    return (ptr -> data.score);
}

template < class T >
bool Stack<T>::Set_Score(const char * player, float scr)
{
    node *ptr;
    ptr = top;
    if(isEmpty())
      return false;
    while(ptr -> data.name != player)
    {
  ptr = ptr -> next;
  if(ptr == NULL)
    return false;
    }
    ptr -> data.score = scr;
    return true;
}

template < class T >
bool Stack<T>::Find_Name(const char * player)
{
    node *ptr;
    ptr = top;
    if(isEmpty())
      return false;
    while(ptr -> data.name != player)
    {
      ptr = ptr -> next;
      if(ptr == NULL)
   return false;
    }
    return true;
}
#endif
```

TaggingSystem.h
Toggle Code#pragma once

```
#include "gmplugin.h"
#include "Stack.h"

class TaggingSystem :
 public Plugin

{

public:
 StringClass ttl[18];
 int scr[17];
 TaggingSystem();
 ~TaggingSystem();


 virtual void OnLoadGlobalINISettings(INIClass *SSGMIni);
 virtual void OnPlayerJoin(int PlayerID,const char *PlayerName);
 virtual void OnLoadLevel();
 virtual void OnGameOver();
 void RecordData();
 void LoadData();
 void TagPlayer(int);
 void doStuff(int);
 void CheckMonth();
 void SetMonth();

private:
 struct  tagData{
     StringClass name;
     float score;
 };
 Stack <tagData> list;
};

int GetMonth();
```

TaggingSystem.cpp
Toggle Code

```
#include "General.h"
#include "TaggingSystem.h"
#include "engine_tt.h"
#include "engine_io.h"
#include "gmgame.h"



int GetMonth()
{
    time_t t = time(0);
```

```
    struct tm * now = localtime( & t );
    return now->tm_mon + 1;
};

TaggingSystem::TaggingSystem()
{
 RegisterEvent(EVENT_GLOBAL_INI,this);
 RegisterEvent(EVENT_PLAYER_JOIN_HOOK,this);
 RegisterEvent(EVENT_LOAD_LEVEL_HOOK,this);
 RegisterEvent(EVENT_GAME_OVER_HOOK,this);
}

TaggingSystem::~TaggingSystem()
{
 UnregisterEvent(EVENT_GLOBAL_INI,this);
 UnregisterEvent(EVENT_PLAYER_JOIN_HOOK,this);
 UnregisterEvent(EVENT_LOAD_LEVEL_HOOK,this);
 UnregisterEvent(EVENT_GAME_OVER_HOOK,this);
}

void TaggingSystem::OnLoadLevel()
{
 LoadData();
 CheckMonth();
 doStuff(2);
}

void TaggingSystem::OnGameOver()
{
 doStuff(1);
 RecordData();
 list.Destroy();
}

void TaggingSystem::SetMonth()
{
 tagData temp_data;
 temp_data.name = "&";
 temp_data.score = float(GetMonth());
 list.Push(temp_data);
}

void TaggingSystem::CheckMonth()
{
 if(list.isEmpty())
    SetMonth();
 else
    if (list.Get_Score("&") != GetMonth())
```

```
    {
      list.Destroy();
      SetMonth();
      doStuff(3);
      Console_Output("%s\n","[TaggingSystem] New Month: Old data has been deleted.");
    }
}

void TaggingSystem::RecordData()
{
 tagData temp_data;
 FILE * pFile;
 pFile =  fopen("scores.txt","w");
 for(int i = 0; i < list.Count(); i++)
 {
    temp_data = list.Peek_At(i);
    fprintf (pFile, "%.2f>%s\n",temp_data.score, temp_data.name.Peek_Buffer());
 }
      fclose (pFile);
}

void TaggingSystem::LoadData()
{

 FILE * pFile;
 pFile = fopen ("scores.txt","r");
 if (!pFile)
    Console_Output("%s\n","[TaggingSystem] Skipped scores.txt");
 else
 {
    char  strc[29],str[29];
    float v;
    tagData temp_data;
    while(!feof(pFile))
    {
        fscanf(pFile, "%f",&v);
  fscanf(pFile, "%s",strc);
  for(int i = 1; i<sizeof(strc)+1 ; i++)
     str[i-1] = strc[i];
  temp_data.name = StringClass(str);
  temp_data.score = v;
  list.Push(temp_data);
    }
    fclose (pFile);
    list.Pop();
 }
}
```

```
void TaggingSystem::doStuff(int choice)
{
 for (SLNode<cPlayer>* PlayerIter = Get_Player_List()->Head(); (PlayerIter != NULL); PlayerIter =
PlayerIter->Next())
 {

      cPlayer *p = PlayerIter->Data();
      StringClass  temp_player;
      if (p->IsActive)
      {
   int ID = p->PlayerId;
   temp_player = StringClass(Find_Player(ID)->PlayerName);
   if(choice == 3)
   {
       tagData temp_data;
       temp_data.name = temp_player;
       temp_data.score = 0;
       list.Push(temp_data);
   }
   else
   {
        if(choice == 1)
        {
    float scr = list.Get_Score(temp_player);
     list.Set_Score(temp_player, scr+=Get_Score(ID));
    }
    else
        if(choice == 2)
     TagPlayer(ID);
   }
  }
 }
}

void TaggingSystem::OnPlayerJoin(int PlayerID,const char *PlayerName)
{
 StringClass player_name=StringClass(Find_Player(PlayerID)->PlayerName);
 tagData temp_data;
       if(list.Find_Name(player_name))
           TagPlayer(PlayerID);
 else
 {
    temp_data.name = player_name;
    temp_data.score=0;
           list.Push(temp_data);
           TagPlayer(PlayerID);
 }
}
```

```
void TaggingSystem::TagPlayer(int ID)
{
 StringClass ws_tag;
 StringClass temp_player = StringClass(Find_Player(ID)->PlayerName);
 float r = list.Get_Score(temp_player);


 if(r <= scr[16])
    ws_tag = ttl[17];
 else
    for(int i = 16; i != 0; i--)
    {
       if(r > scr[i])
             ws_tag = ttl[i];
    }
 Find_Player(ID)->customTag.Format(WideStringClass(ws_tag));
}

void TaggingSystem::OnLoadGlobalINISettings(INIClass *SSGMIni)
{
 ttl[0] = " ";
 SSGMIni->Get_String(ttl[1],"TaggingSystem","tag1","missing data");
 SSGMIni->Get_String(ttl[2],"TaggingSystem","tag2","missing data");
 SSGMIni->Get_String(ttl[3],"TaggingSystem","tag3","missing data");
 SSGMIni->Get_String(ttl[4],"TaggingSystem","tag4","missing data");
 SSGMIni->Get_String(ttl[5],"TaggingSystem","tag5","missing data");
 SSGMIni->Get_String(ttl[6],"TaggingSystem","tag6","missing data");
 SSGMIni->Get_String(ttl[7],"TaggingSystem","tag7","missing data");
 SSGMIni->Get_String(ttl[8],"TaggingSystem","tag8","missing data");
 SSGMIni->Get_String(ttl[9],"TaggingSystem","tag9","missing data");
 SSGMIni->Get_String(ttl[10],"TaggingSystem","tag10","missing data");
 SSGMIni->Get_String(ttl[11],"TaggingSystem","tag11","missing data");
 SSGMIni->Get_String(ttl[12],"TaggingSystem","tag12","missing data");
 SSGMIni->Get_String(ttl[13],"TaggingSystem","tag13","missing data");
 SSGMIni->Get_String(ttl[14],"TaggingSystem","tag14","missing data");
 SSGMIni->Get_String(ttl[15],"TaggingSystem","tag15","missing data");
 SSGMIni->Get_String(ttl[16],"TaggingSystem","tag16","missing data");
 SSGMIni->Get_String(ttl[17],"TaggingSystem","tag17","missing data");

 scr[0] = 0;
 scr[1] = SSGMIni->Get_Int("TaggingSystem","score1", 130000);
 scr[2] = SSGMIni->Get_Int("TaggingSystem","score2", 105000);
 scr[3] = SSGMIni->Get_Int("TaggingSystem","score3", 92000);
 scr[4] = SSGMIni->Get_Int("TaggingSystem","score4", 82000);
 scr[5] = SSGMIni->Get_Int("TaggingSystem","score5", 72000);
 scr[6] = SSGMIni->Get_Int("TaggingSystem","score6", 65000);
 scr[7] = SSGMIni->Get_Int("TaggingSystem","score7", 68000);
```

```
 scr[8] = SSGMIni->Get_Int("TaggingSystem","score8", 51000);
 scr[9] = SSGMIni->Get_Int("TaggingSystem","score9", 44000);
 scr[10] = SSGMIni->Get_Int("TaggingSystem","score10", 37000);
 scr[11] = SSGMIni->Get_Int("TaggingSystem","score11", 30000);
 scr[12] = SSGMIni->Get_Int("TaggingSystem","score12", 23000);
 scr[13] = SSGMIni->Get_Int("TaggingSystem","score13", 17000);
 scr[14] = SSGMIni->Get_Int("TaggingSystem","score14", 11000);
 scr[15] = SSGMIni->Get_Int("TaggingSystem","score15", 5000);
 scr[16] = SSGMIni->Get_Int("TaggingSystem","score16", 2500);

}


TaggingSystem taggingsystem;

extern "C" __declspec(dllexport) Plugin* Plugin_Init()
{
    return &taggingsystem;
}
```

This goes in the ssgm.ini
Toggle Code;Add to the ssgm.ini file.

```
[TaggingSystem]

tag1=General
tag2=Lieutenant General
tag3=Major General
tag4=Brigadier General
tag5=Colonel
tag6=Lieutenant Colonel
tag7=Captain
tag8=First Lieutenant
tag9=Second Lieutenant
tag10=Sergeant Major
tag11=First Sergeant
tag12=Master Sergeant
tag13=Sergeant First Class
tag14=Sergeant
tag15=Corporal
tag16=Private First Class
tag17=Private

score1=130000
score2=105000
score3=92000
score4=82000
SCORE5=72000
```

score6=65000
score7=58000
score8=51000
score9=44000
score10=37000
score11=30000
score12=23000
score13=17000
score14=11000
score15=5000
score16=2500

All the files are included in the *.rar

File Attachments

1) TaggingSystem.rar, downloaded 301 times